

**Phase I SBIR Final Report**  
**Topic: OSD00-HP01**  
**“3-D Voxel-Based Bio-Heat Transfer Code”**

**Security Classification: Unclassified**  
**Contract: N00014-01-M-0071**  
**Data Item No. 0001AD: Final Report**

**Technical: Dr. John Ziriaux / Naval Health Research Center**  
**Program: CDR Stephen Ahlers / Office of Naval Research**  
**Performance Period: Jan 2001 – Jul 2001**

**Submitted by:**

**ThermoAnalytics, Inc.**  
**Principal Investigator: Allen R. Curran**  
**Program Manager: Keith R. Johnson**

**23440 Airpark Blvd**  
**PO Box 66**  
**Calumet, MI 49913**  
**(906) 482-9506**

**July 31, 2001**

The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Navy position, policy, or decision, unless so designated by other documentation.

**20010810 041**

<b>REPORT DOCUMENTATION PAGE</b>			<i>Form Approved</i> <i>OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 31 July 2001		3. REPORT TYPE AND DATES COVERED Final: Jan 2001 – July 2001
4. TITLE AND SUBTITLE 3-D Voxel-Based Bio-Heat Transfer Code			5. FUNDING NUMBERS Contract N00014-01-M-0071	
6. AUTHORS Allen R. Curran, David Nelson, Eric Marttila, Peter Rynes, James Truax				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) ThermoAnalytics, Inc. 23440 Airpark Blvd PO Box 66 Calumet, MI 49913-0066			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research ONR 341, CDR Stephen Ahlers 800 N. Quincy St. Arlington, VA 22217-5660			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Report developed under SBIR contract. The potential health effects of exposure to sources of non-ionizing radio frequency (RF) electro-magnetic energy are an area of continuing interest within the military and civilian communities. Common sources of RF exposure include cellular telephones, magnetic resonance imaging (MRI) systems, and other civilian and military communications and radar systems. One established biological effect of RF exposure is tissue heating; temperature increases of as little as 4°C above normal body temperature can have potentially devastating effects on living tissue. The development of an anatomically realistic thermal computer code, capable of predicting tissue temperatures, is an important accomplishment, both scientifically and for the potential economic benefit deriving from the use of the models. Previous attempts represented the human body as simple, largely homogenous, geometrical shapes that lacked required resolution. Detailed voxel-based anatomical descriptions, such as the Visible Human model, overwhelm general-purpose thermal codes. ThermoAnalytics has further developed its existing commercial thermal software to exploit the structure of these voxel-based descriptions. This prototype thermal code will not be limited to RF heating but will extend to a variety of environmental and therapeutic heating and cooling applications. The Phase I activities are reported here.				
14. SUBJECT TERMS SBIR, thermoregulation, heat transfer, thermal modeling, 3-dimensional modeling, voxel, SAR, FDTD			15. NUMBER OF PAGES 42	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT SAR	

## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
Acknowledgment .....	iii
1.0 IDENTIFICATION OF THE PROBLEM.....	4
1.1 Background.....	5
2.0 PHASE I TECHNICAL OBJECTIVES .....	6
3.0 TISSUE PROPERTIES .....	6
4.0 PHASE I CODE DEVELOPMENT .....	7
4.1 Model Import and Setup .....	7
4.2 Thermal Solution .....	8
4.3 Parallelization .....	14
5.0 VALIDATION.....	17
5.1 Analytical Comparison .....	17
5.2 SAR to Temperature Correlation.....	20
5.3 Qualitative Comparison.....	22
5.4 Monkey Head Test Case Comparison .....	26
5.5 Validation Summary.....	28
6.0 RELEVANT PUBLICATIONS.....	28
Appendix A. Tissue Properties .....	31
Appendix B.1 Class Definitions for the Solver .....	33
Appendix B.2 Solver Source Code.....	36
Appendix B.3 Sample Input File .....	42

### Acknowledgment

This work was supported by the Small Business Innovation Research (SBIR) Program through the Office of the Secretary of Defense (OSD) and administered through the Office of Naval Research (ONR). The Program Officer was CDR Stephen Ahlers of ONR and the Technical Monitor was John Ziriaux of Naval Health Research Center at Brooks AFB. This Final Report is Data Item No. 0001AD of the Contract Data Requirements List for Contract N00014-01-M-0071.



## 1.0 IDENTIFICATION OF THE PROBLEM

The potential health effects of exposure to sources of non-ionizing radio frequency (RF) electromagnetic energy are an area of continuing interest within the military and civilian communities. Common sources of RF exposure include mobile and cellular telephones, magnetic resonance imaging (MRI) systems, wireless local area networks, transmitting antennae, and other civilian and military communications and radar systems. One established biological effect of RF exposure is tissue heating; temperature increases as little as 4°C above normal body temperature can have potentially devastating effects on living tissue [31]. This report discusses the development of a prototype of an anatomically realistic thermal computer code capable of predicting the thermal effects of known RF sources on human and animal tissue.

The Institute of Electrical and Electronics Engineers (IEEE) recently adopted revised standards for human exposure to RF fields, specifying maximum permissible exposure (MPE) levels. The MPE level, for a given exposure frequency, is stated in terms of the specific absorption rate (SAR) – the rate of energy absorption per unit mass of tissue. The MPE depends upon the duration of exposure and whether the exposure is “controlled” to avoid physiological hazards [28]. The rate of temperature increase in tissue is a function of the specific absorption rate. The local rate of temperature increase in a homogeneous *phantom model* is approximately linear with the SAR, for short-duration exposures [27]. Phantom models however, are of little value in predicting tissue heating effects of RF exposure in humans or animals. The effects of metabolic heating, blood flow, and tissue heterogeneity complicate the relationship between the rate of temperature increase and the local SAR, such that a simple linear model is not generally appropriate.

The anatomical complexity and high degree of spatial resolution required to capture predicted SAR “hot spots” have led to the development of realistic anatomical data sets [1,2]. These *voxel*-based descriptions are created by dividing the space occupied by the object being described (e.g., the human body) into a three dimensional grid of small, equal-sized volume elements known as voxels. By definition, each voxel can contain only one type of tissue; a very large number of small voxels (e.g., cubes with 1mm edges) is therefore needed to capture enough detail to accurately predict SARs. The properties of the tissue in each voxel and the predicted SARs constitute the input to a thermal model, which then predicts tissue temperatures. Consequently, general-purpose off-the-shelf thermal codes are quickly overwhelmed by the size of thermal problems that are derived from voxel-based anatomical descriptions. ThermoAnalytics proposed to develop new thermal software to exploit the simple structure of the voxel-based description. The resulting thermal code is compact, accurate, and able to accommodate extremely large data sets as input.

This report addresses the issue of thermoregulation and bio-heat transfer modeling as it pertains to complex voxel-based anatomical descriptions with thermal loads from electromagnetic (EM) irradiation. The level and location of heat loading are derived using the finite difference time domain (FDTD) method and applied to the model as a specific absorption rate. The applicability of the resulting computer code is not limited to RF heating however, but extends to a variety of environmental and therapeutic heating and cooling applications.



## 1.1 Background

The development of accurate, whole-body, predictive models of heat transfer in humans and primates would be an extremely important accomplishment, both scientifically and for the potential economic benefit deriving from the use of the models. While whole-body models have been described in the literature [2,10], the lack of anatomical detail and precision, and limited thermoregulatory mechanisms, means their applications are limited to describing relatively gross thermal response.

Thermoregulation is an ensemble of physiological processes that differ among species, and which collectively complicate the development of predictive models. Conduction, convection, and radiation heat transfer, though well understood and consistently modeled with high levels of accuracy, are complicated by evaporation (sweating, respiration) and other phase change processes, as well as physiological responses such as vasomotor reflexes. Incomplete anatomical data sets have also prevented thermoregulatory models from providing realistic output. Furthermore, accurate model validations are difficult to perform because experiments are invasive and boundary conditions can be difficult to quantify. Animal models are useful, but size and morphological differences make it extremely difficult to extrapolate from animals to humans. Extrapolation of experimental results from animal models to humans is further complicated by species differences in thermal dissipation mechanisms, notably evaporative cooling. The thermoregulatory system of non-human primate species is somewhat similar to that of humans, providing better opportunities for experimental studies. Primate cooling physiology, particularly their ability to sweat, differs from that of humans however [30]. For this reason, primate studies, though very useful, have somewhat limited application [12,29].

Despite these factors, which thus far have prevented the development of accurate anatomical human models, mathematical expressions have been derived which, though conceptually simple, accurately describe local temperature responses of homogeneous tissues to thermal loads. Temperature predictions are commonly based on Pennes' bio-heat transfer equation [11], which is the conduction heat equation with added terms to represent the metabolic heat load and convective effect of local blood flow:

$$k_t \nabla^2 T + \rho_b c_b (T_{art} - T) + \dot{q}_m = \rho_t c_t \frac{\partial T}{\partial t} \quad (1)$$

The left-hand side terms of equation 1 represent, respectively, conduction through tissue with thermal conductivity  $k_t$ , heat convection associated with blood flow  $\rho_b c_b$ , and metabolic heating. The right-hand side term describes the rate of energy increase of the tissue volume.

Although the theoretical basis of the Pennes' model is controversial, it has been validated in numerous species and tissues and is widely accepted as an appropriate bio-heat transfer equation for most applications [2,10]. It has also been shown that blood flow, vasodilatation and constriction, and other circulatory processes can be successfully modeled [6], as well as metabolic effects and the role of blood chemistry [10]. Modern predictive techniques and powerful high-speed computers make application of the bio-heat equation feasible on animal and human anatomical subjects. Data sets, particularly the voxelized data available through the Visible Human Project [National Library of Medicine] and others provide an anatomical description detailed enough to apply the bio-heat equation on a scale necessary to take into



account the effects of differing tissue properties, vasomotor and sudomotor responses, and the convective effects of the circulatory system.

## 2.0 PHASE I TECHNICAL OBJECTIVES

The main objective of the Phase I work was to develop a prototype for an anatomically realistic thermal modeling tool. The prototype, written in C++, runs on both Unix and Windows systems and is able to:

1. Read voxel-based digital anatomical data sets (.raw files and the corresponding tissue.txt file) and produce a thermal nodal network, based on Pennes' bio-heat transfer equation, in which each thermal node corresponds to one voxel,
2. Use known thermal properties of different tissues and materials to determine the thermal capacitance of each node and the conduction between nodes,
3. Read and apply the specific absorption rate (SAR), as predicted by the FDTD program, to each thermal node for a period of time set by the user,
4. Solve for the time varying temperature of each node using finite difference techniques,
5. Produce three dimensional temperature files, of the same format as the SAR file, at a specified simulation time, and,
6. Has been validated by comparison to analytical test cases and experimental measurements.

The prototype code includes terms for metabolic heating and blood flow in the nodal energy budget, although these will not be available to the user until Phase II. Our Phase I activities also included generating a database of the thermal properties of different tissues and relevant materials. Thermal predictions were validated using analytic test cases and measured data, obtained from the Directed Energy BioEffects Laboratory (DEBL) at Brooks AFB, taken on a Rhesus monkey carcass subjected to far-field irradiation under known conditions of power density, frequency, and orientation.

## 3.0 TISSUE PROPERTIES

Thermal properties (thermal conductivity, specific heat, mass density, thermal diffusivity) have been identified for virtually all the relevant tissues. In some cases, there are multiple entries corresponding to different species (rat, human, monkey). [Appendix A]

The metabolic heat production and blood flow per unit tissue volume are determined from the literature, where available, for most metabolically active tissues. In some cases, the metabolic heat production (per unit tissue volume) is estimated from O<sub>2</sub> consumption rates.

For many tissues, blood flow rates are tabulated as a range of values. This is particularly relevant to superficial tissues that exhibit significant vasoconstriction (or vasodilatation) according to thermal state. Modeling this thermoregulatory response will be an important aspect of Phase II work.





## 4.0 PHASE I CODE DEVELOPMENT

In this section we will describe the development of the Bio-Heat Transfer software delivered at the end of Phase I. Originally, this prototype was to be based on ThermoAnalytics' existing thermal modeling software, RadTherm. Based on feedback from the kick-off meeting, an emphasis was placed on producing a prototype thermal solver that included parallelized code and for which the source code could be made widely available. For this reason, the voxel thermal solver was implemented as a sub-solution type of our main solver (similar to the tri-diagonal solver we use for 1D terrain models) in such a way that it can also be made standalone.

### 4.1 *Model Import and Setup*

The modeling process begins by importing voxel data sets containing tissue types and SAR values. The large size of voxel data sets (e.g., over 45 million voxels for the "2mm man," a 293 x 170 x 939 voxel representation of a human body) requires that we utilize compact data structures, avoid duplicate data representations, and eliminate storage by recalculating simple quantities as they are needed. Fortunately, the simple structure of the voxel description lends itself to this type of optimization. The C++ class (See Appendix B.1) that holds voxel data consists of variables that hold the x, y, z dimensions of each voxel, the number of voxels in the x, y, z directions ( $n_x$ ,  $n_y$ ,  $n_z$ ), a one-dimensional array dynamically sized to hold  $n_x \cdot n_y \cdot n_z$  instances of a voxel data class, and functions for accessing the voxel information at a given index or position. The voxel data class contains storage for temperature, temperature from the last time step, the SAR heating value, and the tissue type of the voxel. This storage scheme is several times smaller than if voxel data were stored using the data structures typically defined for solid elements. (The storage for a single solid element consists of 6 integers that hold vertex indices. There are roughly as many vertices as there are solid elements and each requires 3 floats to hold the x, y, z coordinates.)

Several parameters that control the thermal solution are required and these are read from a text file prior to invoking the thermal solution. These parameters are listed in Table 1, next page. A sample text input file is included in Appendix B.3.



Table 1. Model Parameter Input File

Parameter	Units
Number of voxels in x, y, and z dimensions	n/a
Voxel size in x, y, and z dimensions	mm
Duration of exposure to RF heating	Seconds
Duration of thermal simulation	Seconds
Power density of RF heating	mW/cm <sup>2</sup>
Initial temperature of the object	°C
Time step of thermal simulation	Seconds
Ambient temperature	°C
Surface convection coefficient	W/m <sup>2</sup> -K

## 4.2 Thermal Solution

The thermal solver for the Phase I prototype solves a variation of Pennes' bio-heat equation which includes the effects of local blood flow, metabolic heating, convective heat transfer, and the heating effects of RF energy. The following equation, repeated from Section 1.1 with added terms  $\dot{q}_{sar}$  for RF heating and  $\dot{q}_{conv}$  for convective heat transfer, is:

$$k_t \nabla^2 T + \rho_b c_b (T_b - T) + \dot{q}_m + \dot{q}_{conv} + \dot{q}_{sar} = \rho_t C_{p_t} \frac{\partial T}{\partial t} \quad (2)$$

The spatial derivatives in Equation 2 are discretized for the voxelized domain using second order finite difference expressions, and the Crank-Nicholson scheme is applied for discretization in the time domain. The discretization yields the following equation:





$$\begin{aligned}
 & k_t \left[ \frac{T^{n+1}_{i-1,j,k} - 2T^{n+1}_{i,j,k} + T^{n+1}_{i+1,j,k}}{2\Delta x^2} + \frac{T^n_{i-1,j,k} - 2T^n_{i,j,k} + T^n_{i+1,j,k}}{2\Delta x^2} \right] + \\
 & k_t \left[ \frac{T^{n+1}_{i,j-1,k} - 2T^{n+1}_{i,j,k} + T^{n+1}_{i,j+1,k}}{2\Delta y^2} + \frac{T^n_{i,j-1,k} - 2T^n_{i,j,k} + T^n_{i,j+1,k}}{2\Delta y^2} \right] + \\
 & k_t \left[ \frac{T^{n+1}_{i,j,k-1} - 2T^{n+1}_{i,j,k} + T^{n+1}_{i,j,k+1}}{2\Delta z^2} + \frac{T^n_{i,j,k-1} - 2T^n_{i,j,k} + T^n_{i,j,k+1}}{2\Delta z^2} \right] + \\
 & n_b C_{p_b} \left[ \frac{T^{n+1}_{b,i,j,k} - T^{n+1}_{i,j,k}}{2} + \frac{T^n_{b,i,j,k} - T^n_{i,j,k}}{2} \right] + \\
 & \frac{h_s A_s}{\Delta x \Delta y \Delta z} \left[ \frac{T^{n+1}_\infty - T^{n+1}_{i,j,k}}{2} + \frac{T^n_\infty - T^n_{i,j,k}}{2} \right] + \\
 & \phi^{n+1}_{m_{i,j,k}} + \phi^{n+1}_{sar_{i,j,k}} = \rho_t C_{p_t} \frac{T^{n+1}_{i,j,k} - T^n_{i,j,k}}{\Delta t}
 \end{aligned} \tag{3}$$

The superscripts  $n$  and  $n+1$  indicate time step  $n$  and time step  $n+1$ , respectively. The parameter  $A_s$  is the surface area of a voxel face that is exposed to ambient air, and  $h_s$  is the coefficient for convective heat transfer between exposed voxel faces and the external air. The parameters  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  denote the dimensions of an individual voxel in the x, y and z directions, and  $\Delta t$  is the time step interval.

Equation 3 will be rearranged to solve for  $T^{n+1}_{i,j,k}$ , and similar terms will be combined for more efficient numerical evaluation. The first step is to separate the terms that contain temperatures from the  $n+1$  time step:



$$\begin{aligned}
 & \frac{k_t}{\Delta x^2} [T^{n+1}_{i-1,j,k} - T^{n+1}_{i,j,k} + T^{n+1}_{i+1,j,k} - T^{n+1}_{i,j,k}] + \\
 & \frac{k_t}{\Delta x^2} [T^n_{i-1,j,k} - T^n_{i,j,k} + T^n_{i+1,j,k} - T^n_{i,j,k}] + \\
 & \frac{k_t}{\Delta y^2} [T^{n+1}_{i,j-1,k} - T^{n+1}_{i,j,k} + T^{n+1}_{i,j+1,k} - T^{n+1}_{i,j,k}] + \\
 & \frac{k_t}{\Delta y^2} [T^n_{i,j-1,k} - T^n_{i,j,k} + T^n_{i,j+1,k} - T^n_{i,j,k}] + \\
 & \frac{k_t}{\Delta z^2} [T^{n+1}_{i,j,k-1} - T^{n+1}_{i,j,k} + T^{n+1}_{i,j,k+1} - T^{n+1}_{i,j,k}] + \\
 & \frac{k_t}{\Delta z^2} [T^n_{i,j,k-1} - T^n_{i,j,k} + T^n_{i,j,k+1} - T^n_{i,j,k}] + \\
 & m_b C_{p_b} [T^{n+1}_{i,j,k} - T^{n+1}_{i,j,k}] + m_b C_{p_b} [T^n_{i,j,k} - T^n_{i,j,k}] \\
 & \frac{h_s A_s}{\Delta x \Delta y \Delta z} [T^{n+1}_{\infty} - T^{n+1}_{i,j,k}] + \frac{h_s A_s}{\Delta x \Delta y \Delta z} [T^n_{\infty} - T^n_{i,j,k}] \\
 & 2\phi^{n+1}_{m_{i,j,k}} + 2\phi^{n+1}_{sa_{i,j,k}} = 2\rho_l C_{p_l} \frac{T^{n+1}_{i,j,k} - T^n_{i,j,k}}{\Delta t}
 \end{aligned} \tag{4}$$

Collecting the terms in Equation 4 that contain  $T^{n+1}_{i,j,k}$  and  $T^n_{i,j,k}$  yields:

$$\begin{aligned}
 & -T^{n+1}_{i,j,k} \left[ \frac{2k_t}{\Delta x^2} + \frac{2k_t}{\Delta x^2} + \frac{2k_t}{\Delta y^2} + m_b C_{p_b} + \frac{h_{s_{i,j,k}} A_{s_{i,j,k}}}{\Delta x \Delta y \Delta z} + \frac{2\rho_l C_{p_l}}{\Delta t} \right] + \\
 & -T^n_{i,j,k} \left[ \frac{2k_t}{\Delta x^2} + \frac{2k_t}{\Delta x^2} + \frac{2k_t}{\Delta y^2} + m_b C_{p_b} + \frac{h_{s_{i,j,k}} A_{s_{i,j,k}}}{\Delta x \Delta y \Delta z} - \frac{2\rho_l C_{p_l}}{\Delta t} \right] + \\
 & \frac{k_t}{\Delta x^2} [T^{n+1}_{i-1,j,k} + T^{n+1}_{i+1,j,k}] + \frac{k_t}{\Delta x^2} [T^n_{i-1,j,k} + T^n_{i+1,j,k}] + \\
 & \frac{k_t}{\Delta y^2} [T^{n+1}_{i,j-1,k} + T^{n+1}_{i,j+1,k}] + \frac{k_t}{\Delta y^2} [T^n_{i,j-1,k} + T^n_{i,j+1,k}] + \\
 & \frac{k_t}{\Delta z^2} [T^{n+1}_{i,j,k-1} + T^{n+1}_{i,j,k+1}] + \frac{k_t}{\Delta z^2} [T^n_{i,j,k-1} + T^n_{i,j,k+1}] + \\
 & m_b C_{p_b} [T^{n+1}_{i,j,k}] + m_b C_{p_b} [T^n_{i,j,k}] + \\
 & \frac{h_s A_s}{\Delta x \Delta y \Delta z} [T^{n+1}_{\infty}] + \frac{h_s A_s}{\Delta x \Delta y \Delta z} [T^n_{\infty}] + \\
 & 2\phi^{n+1}_{m_{i,j,k}} + 2\phi^{n+1}_{sa_{i,j,k}} = 0
 \end{aligned} \tag{5}$$

Equation 5 is rearranged to prepare for solving for  $T^{n+1}_{i,j,k}$  :



$$\begin{aligned}
 T^{n+1}_{i,j,k} & \left[ \frac{2k_l}{\Delta x^2} + \frac{2k_l}{\Delta x^2} + \frac{2k_l}{\Delta y^2} + n_b C_{p_b} + \frac{h_{s,j,k} A_{s,j,k}}{\Delta x \Delta y \Delta z} + \frac{2\rho_l C_{p_l}}{\Delta t} \right] = \\
 & -T^n_{i,j,k} \left[ \frac{2k_l}{\Delta x^2} + \frac{2k_l}{\Delta x^2} + \frac{2k_l}{\Delta y^2} + n_b C_{p_b} + \frac{h_{s,j,k} A_{s,j,k}}{\Delta x \Delta y \Delta z} - \frac{2\rho_l C_{p_l}}{\Delta t} \right] + \\
 & \frac{k_l}{\Delta x^2} [T^{n+1}_{i-1,j,k} + T^{n+1}_{i+1,j,k}] + \frac{k_l}{\Delta x^2} [T^n_{i-1,j,k} + T^n_{i+1,j,k}] + \\
 & \frac{k_l}{\Delta y^2} [T^{n+1}_{i,j-1,k} + T^{n+1}_{i,j+1,k}] + \frac{k_l}{\Delta y^2} [T^n_{i,j-1,k} + T^n_{i,j+1,k}] + \\
 & \frac{k_l}{\Delta z^2} [T^{n+1}_{i,j,k-1} + T^{n+1}_{i,j,k+1}] + \frac{k_l}{\Delta z^2} [T^n_{i,j,k-1} + T^n_{i,j,k+1}] + \\
 & n_b C_{p_b} [T^{n+1}_{i,j,k}] + n_b C_{p_b} [T^n_{i,j,k}] + \\
 & \frac{h_s A_s}{\Delta x \Delta y \Delta z} [T^{n+1}_{\infty}] + \frac{h_s A_s}{\Delta x \Delta y \Delta z} [T^n_{\infty}] + \\
 & 2\phi^{n+1}_{m,i,j,k} + 2\phi^{n+1}_{sar,i,j,k}
 \end{aligned} \tag{6}$$

Equation 6 can be simplified by defining a parameter  $C$ , which takes on several different values depending on the position of the adjacent voxels, the presence of blood flow, and whether or not any of the voxel faces are exposed to convection with ambient air. The expressions for evaluating the parameter are as follows<sup>1</sup>:

<sup>1</sup> The parameter  $C$  must be modified for the conduction terms when the adjacent voxel has a different thermal conductivity than the current voxel. In this case the parameter takes the form of

$C = \frac{k_{i1} k_{i2}}{(k_{i1} + k_{i2}) \Delta x^2}$ , where  $k_{i1}$  is the thermal conductivity of the current voxel, and  $k_{i2}$  is the thermal conductivity of the adjacent voxel.



$$\begin{aligned}
 C &= \frac{k_i}{\Delta x^2} \text{ (for conduction terms with subscript } i \pm 1) \\
 C &= \frac{k_i}{\Delta y^2} \text{ (for conduction terms with subscript } j \pm 1) \\
 C &= \frac{k_i}{\Delta z^2} \text{ (for conduction terms with subscript } k \pm 1) \\
 C &= \rho_b C_{p_b} \text{ (for blood flow terms)} \\
 C &= \frac{h_s}{\Delta x} \text{ (for convection on surfaces in the } y-z \text{ plane)} \\
 C &= \frac{h_s}{\Delta y} \text{ (for convection on surfaces in the } x-z \text{ plane)} \\
 C &= \frac{h_s}{\Delta z} \text{ (for convection on surfaces in the } x-y \text{ plane)}
 \end{aligned} \tag{7}$$

Applying the expressions in (7) to Equation 6 yields:

$$\begin{aligned}
 T^{n+1}_{i,j,k} \left[ \sum_{l=1}^N C_l + \frac{2\rho_l C_{p_l}}{\Delta t} \right] &= -T^n_{i,j,k} \left[ \sum_{l=1}^N C_l - \frac{2\rho_l C_{p_l}}{\Delta t} \right] + \sum_{l=1}^N C_l T_l^{n+1} + \sum_{l=1}^N C_l T_l^n + \\
 &2\phi^{n+1}_{m_{i,j,k}} + 2\phi^{n+1}_{sar_{i,j,k}},
 \end{aligned} \tag{8}$$

where  $l$  is an index for the summation of all parameters  $C$  that apply to voxel  $i,j,k$ . The limit  $N$  is the total number of  $C$  parameters that apply to voxel  $i,j,k$ , and  $T_l$  is the temperature associated with parameter  $C_l$ , which could be the adjacent voxel temperature, ambient air temperature, or blood temperature. Solving Equation 8 for  $T^{n+1}_{i,j,k}$  and rearranging some terms leads to the equation that is encoded in the numerical solution:

$$T^{n+1}_{i,j,k} = \frac{1}{\sum_{l=1}^N C_l + \frac{2\rho_l C_{p_l}}{\Delta t}} \left\{ \sum_{l=1}^N C_l T_l^{n+1} + \sum_{l=1}^N C_l T_l^n - T^n_{i,j,k} \left[ \sum_{l=1}^N C_l - \frac{2\rho_l C_{p_l}}{\Delta t} \right] + 2\phi^{n+1}_{m_{i,j,k}} + 2\phi^{n+1}_{sar_{i,j,k}} \right\} \tag{9}$$

Equation 9 is applied for each voxel in the domain and the resulting set of equations is solved using an iterative successive over-relaxation solver. The source code for a portion of the voxel thermal solver is included in Appendix B.2.

As mentioned earlier, the voxel thermal solver is implemented as a sub-solution to the main thermal solver which is in RadTherm. This was done by modifying the main thermal solver to allow the addition of an arbitrary number of voxelized domains. During each iteration of the main thermal solver, a single iteration of the voxel thermal solver is performed. This implementation allows for the reuse of much of the main thermal solver architecture, and will also make it easier to add radiative heat transfer boundary conditions in the future.



# Thermoregulation Solver

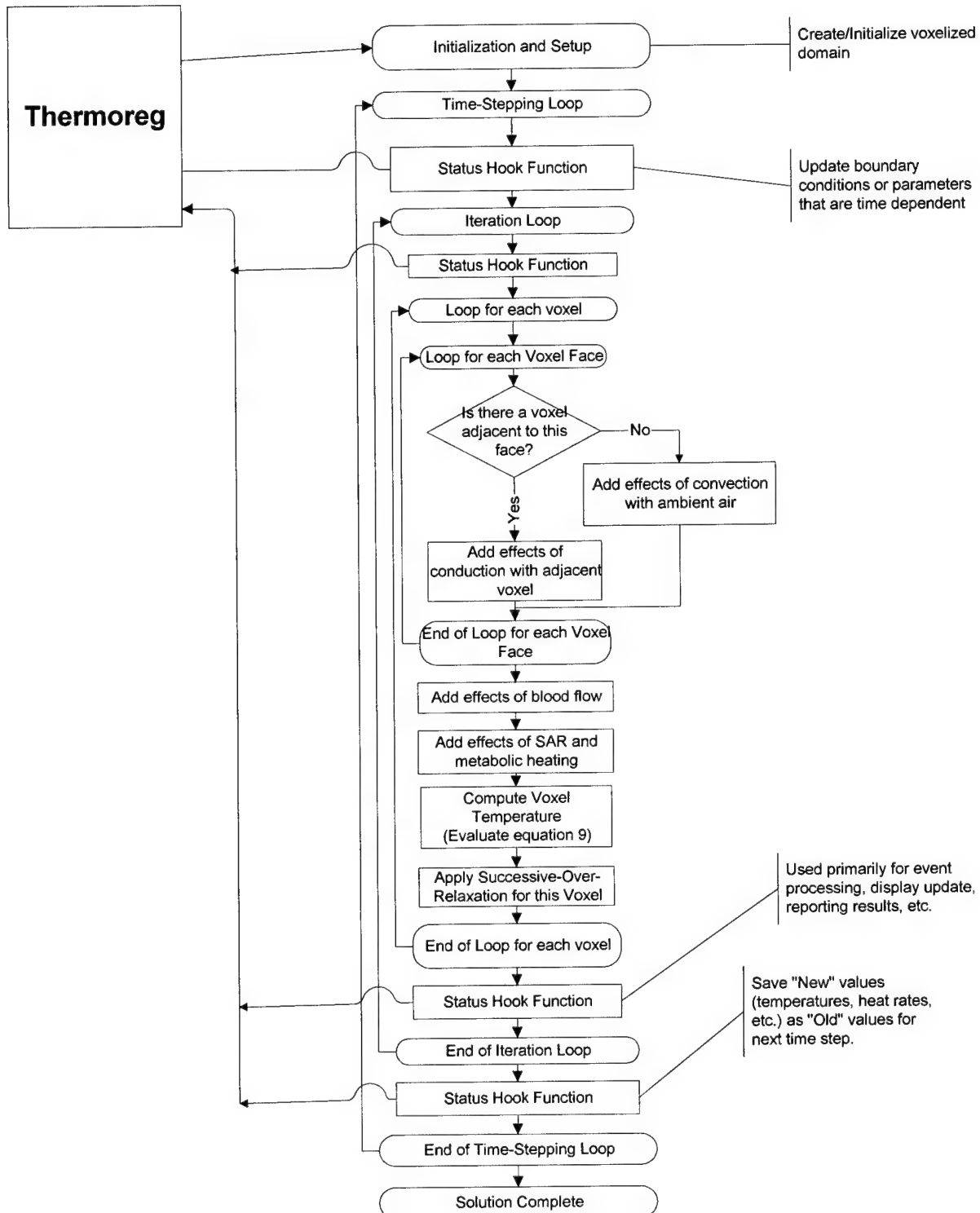


Figure 1: Flowchart of Thermal Solver



Using the solution parameters as specified in Table 1, the thermal solver initializes each voxel using the initial object temperature. The solution proceeds with solving for the voxel temperatures (as described above) at each time step for the specified duration. The normalized RF heating values are multiplied by the specified power density and applied to each voxel from the beginning of the solution up to the duration of exposure. After this time period is over, the RF heating is disabled. A constant convection boundary condition is applied at each voxel surface that is exposed to external air, using the specified convection coefficient and ambient air temperature. After the thermal solution is complete, a dataset containing the temperature of each voxel is written to a binary file in the same format as the SAR heating values. Figure 1 contains a flowchart of the solver.

### 4.3 Parallelization

The objective for parallelizing the voxel-based thermal solver was to be able to solve large, i.e., memory intensive, problems on a distributed memory computer cluster. We utilized the current standard library for writing distributed memory parallel programs, the Message Passing Interface (MPI). After reviewing the literature regarding the parallelization of various iterative solution techniques, we chose to use Successive Over-Relaxation (SOR) as our solution method, primarily because it requires the least amount of storage. All of the solution methods considered, including those that generally provide faster convergence, e.g., *multigrid* and FFT, require storage that is linear with the number of nodes. However, the amount of storage required per node for SOR is a fraction of that required by the other techniques.

We followed the typical approach of partitioning the thermal nodes among the parallel processors to minimize inter-processor communication. Since in most test subjects, the x-y cross section is the smallest, the problem domain was partitioned along the z-axis. Each partition contains one or two extra x-y “slices” of thermal nodes, which are a copy of the nodes whose temperatures are calculated by the adjacent processor(s). Since the temperatures of these slices are only updated at the end of each iteration, the data dependencies across processor boundaries are therefore changed from “Gauss-Seidel style” (in which the results from the current iteration are used as they become available) to “Jacobi style” (in which the results from the previous iteration are used throughout the current iteration). The temperatures of boundary nodes are exchanged between processors via point-to-point communication. Figures 2 and 3 summarize the communication between processes.

In order to improve scalability, the data needed to solve the entire thermal problem (i.e., tissue IDs, temperatures, and SAR values) are never stored on a single processor. Instead, processor 0 reads (writes) buffer-fulls of information that is scattered to (gathered from) the other processors. In the case of the 2mm man model, in which the data is too large to be contained in a single process (under RedHat 7.1), the parallel version of the code can be used to run the model in two processes running on a single CPU.

We installed the LAM 6.5.3 version (free Notre Dame version) of MPI on our network of PCs running Linux (5 Intel P-III 1000Mhz PCs with 512Mb to 1Gb of physical memory each and 2 AMD 1333Mhz PCs with 784Mb of physical memory). Typical performance for the serial and parallel version of the code is summarized in Table 2 and Figure 4.

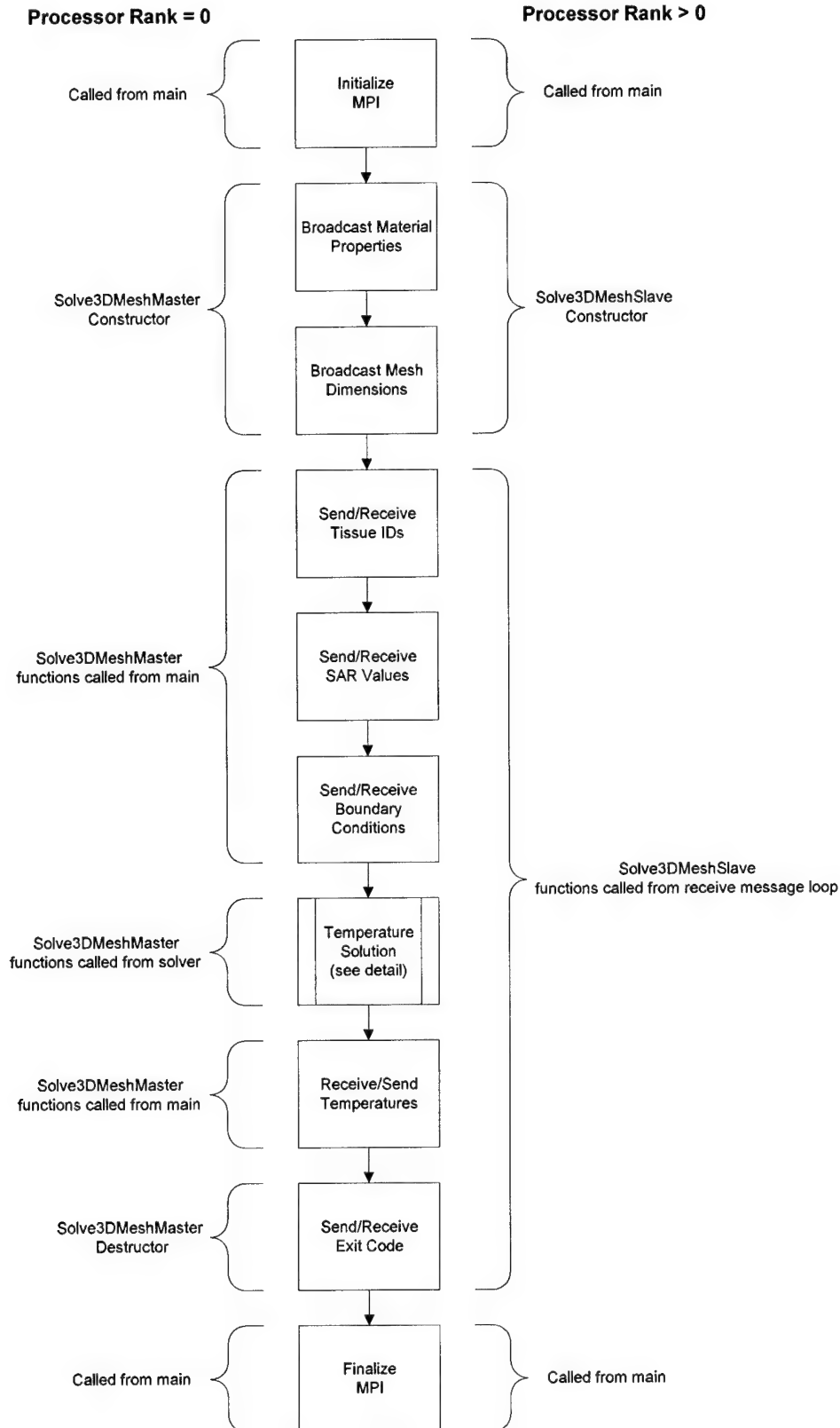
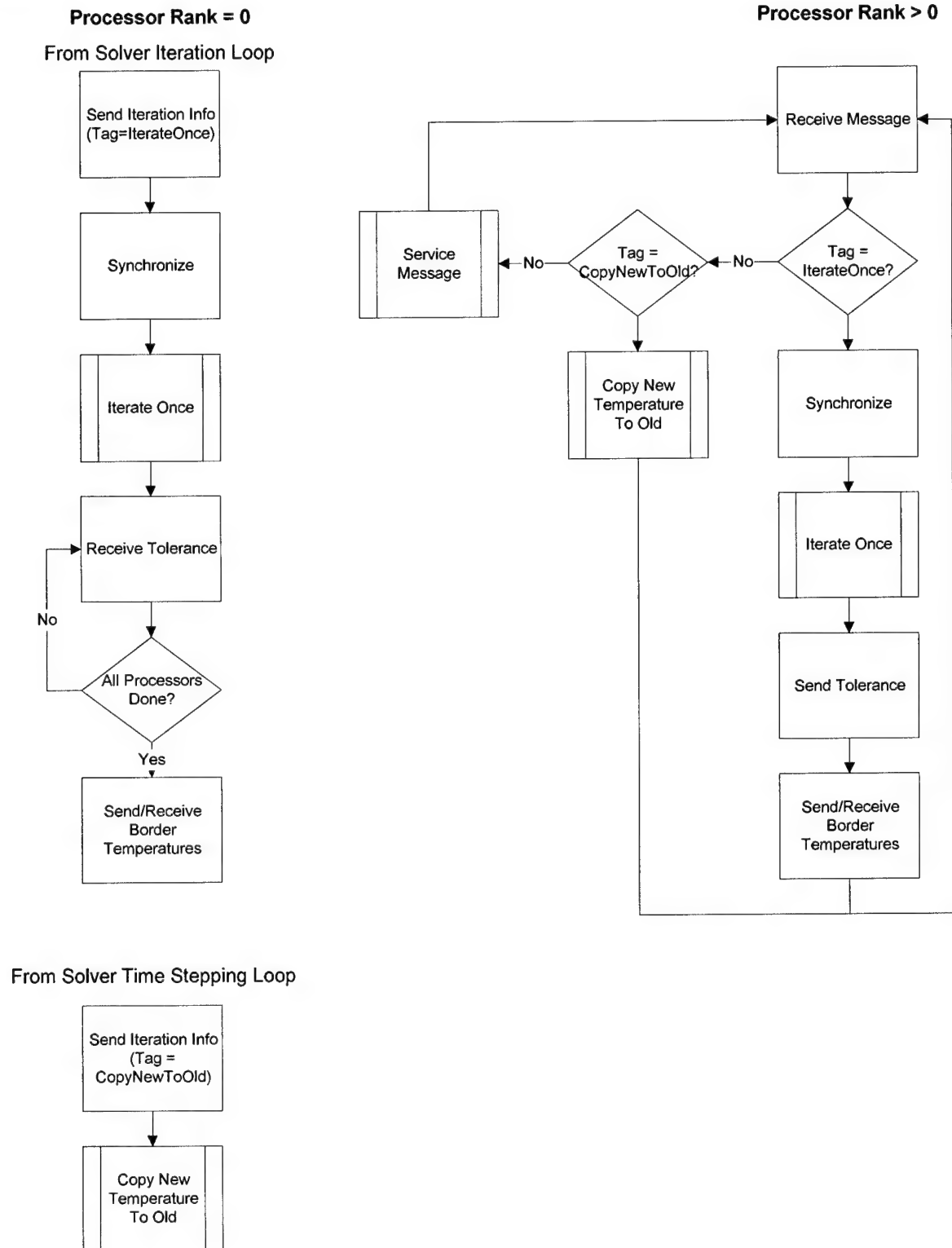


Figure 2: Overview of Communication within the Parallel Version of the Code





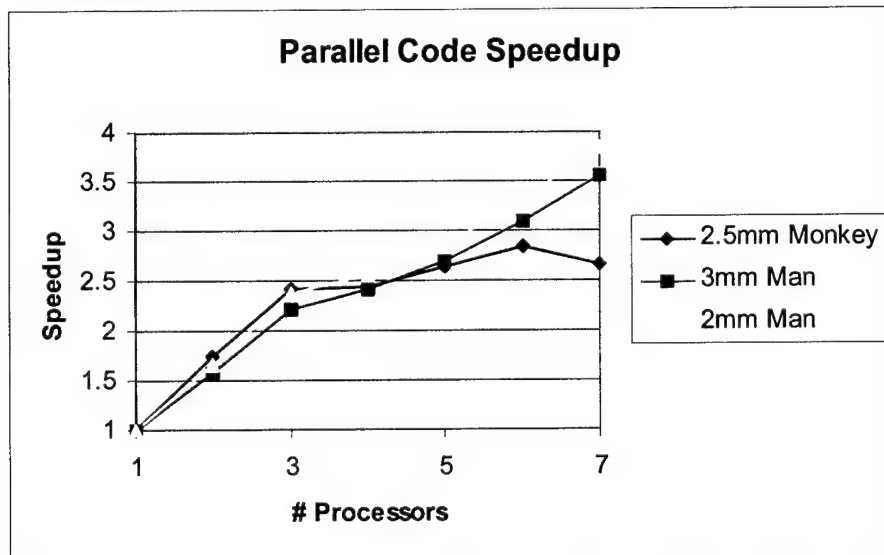
**Figure 3: Parallel Solver Communication during Temperature Solution**



**Table 2: Typical runtimes for 30 seconds of simulation using a 5 second time step on a single CPU.**

Model	# Voxels	# Non-Air Voxels	Runtime (seconds)	Runtime (seconds)
			PIII-1000	AMD-1333
2.5mm Monkey	2,281,600	397,181	73	45
3mm Man	13,987,344	3,875,766	637	393
2mm Man	46,771,590	13,078,893	2355	2011*

\*Not enough RAM to hold entire model.



**Figure 4: Typical speedup for the parallel version of the code. The first five processors were PIII-1000s; the last two were AMD-1333s. All processes fit in available RAM.**

**Note:** the 2mm Man model was too large to run in one process; for this model two processes were running on a single CPU.

## 5.0 VALIDATION

The Phase I Code has been tested against measured temperatures under controlled conditions and against analytically derived data. In this section we will describe the process and the results of the validation.

### 5.1 Analytical Comparison

The analytical test case involved immersing a homogeneous isothermal sphere into an infinite water bath and then allowing it to reach equilibrium temperature.



**Table 3: Analytical Test Case**

Sphere Properties	Initial Conditions	Boundary Conditions
$\rho = 3000 \text{ kg/m}^3$ $k = 20 \text{ W/m K}$ $c_p = 1001.0 \text{ J/kg K}$	$T_{i, \text{sphere}} = 335 \text{ }^\circ\text{C}$ $T_\infty = 20 \text{ }^\circ\text{C}$	convection, $h = 6000 \text{ W/m}^2 \text{ K}$ no radiation

The exact solution for the transient, one-dimensional application of the heat equation for a sphere, in dimensionless form, is;

$$\theta^* = \sum_1^\infty C_n \exp(-\xi_n^2 Fo) \frac{1}{\xi_n r^*} \sin(\xi_n r^*)$$

where;

$$Fo = \alpha \tau / r_o^2$$

$$C_n = \frac{4[\sin(\xi_n) - \xi_n \cos(\xi_n)]}{2\xi_n - \sin(2\xi_n)}$$

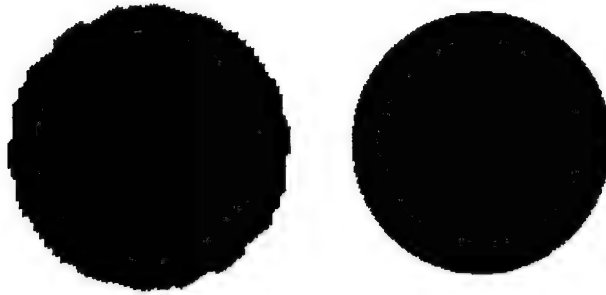
$$\theta^* = \frac{T_{\text{sphere}} - T_\infty}{T_{\text{initial, sphere}} - T_\infty}$$

$\alpha$  is the thermal diffusivity of the sphere material,  $\tau$  is the simulation time,  $r_o$  is the sphere radius,  $r^* \equiv r/r_o$ ,  $Fo$  is the Fourier Number,  $T_\infty$  is the constant water bath temperature and the values of  $\xi_n$  are the eigenvectors to the equation;

$$1 - \xi_n \cot \xi_n = Bi$$

where  $Bi$  is the Biot number [34,35].

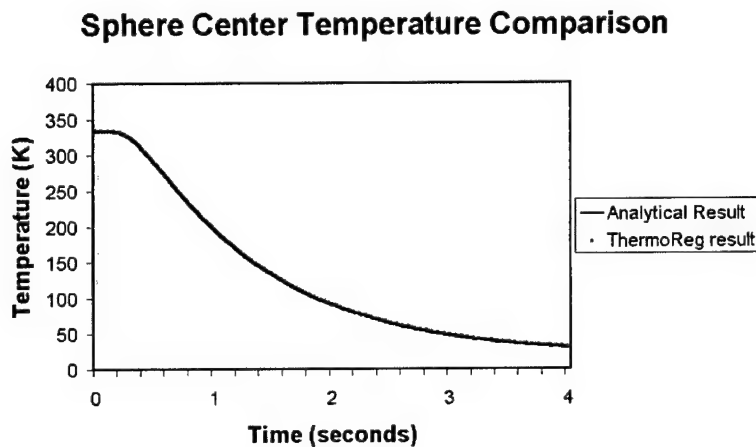
An inherent trait of voxel geometry is that objects are modeled in Cartesian space. Spheres and all other non-planar surfaces, modeled with voxels, must be approximated by cubes, as shown in Figure 5.



**Figure 5: Spheres modeled in Cartesian and Spherical coordinates**

A result of this approximation is that error is introduced into the simulation of a non-planar surface, since the modeled surface area of the volume is not exact. This error causes a slightly non-uniform transient temperature distribution to be calculated over the surface of the sphere.

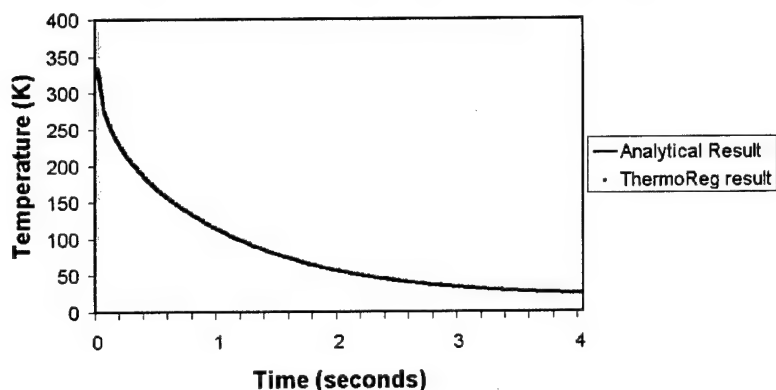
We compared data points at the center and at the surface of the sphere. Figure 6 shows the temperature comparison at the center of the sphere. The edge temperature plot shown in Figure 7 is averaged over the sphere surface in order to account for the voxel geometry induced error.



**Figure 6: Sphere Temperature Comparison, sphere center**



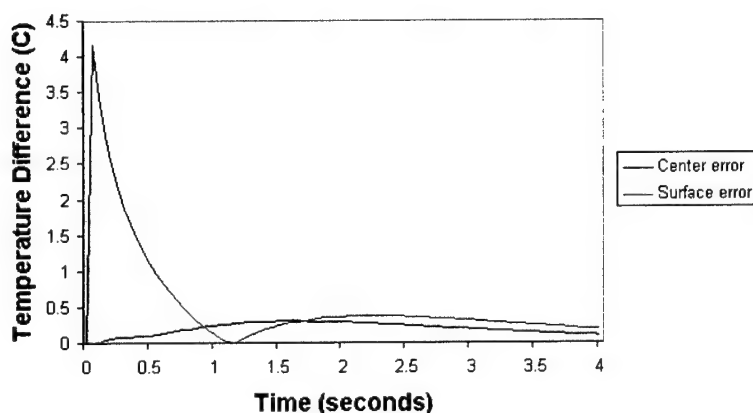
### Sphere Edge Temperature Comparison



**Figure 7: Sphere Temperature Comparison, sphere surface**

The error between the Phase I Code prediction and the analytical solution is plotted in Figure 8. Considering the large initial temperature, the rapid rate of temperature change, and the voxel approximation to a sphere, the error is considered to be negligible.

### Analytical Comparison, Error



**Figure 8: Analytical Comparison Error**

## 5.2 SAR to Temperature Correlation

The locations of thermal “hot spots” within the voxel models correspond to the locations of the peak specific absorption rate (SAR), as predicted by the FDTD code. The rate of tissue temperature increase can be predicted (approximately) for short-term exposures by a simple extrapolation. For an adiabatic system, with negligible thermal diffusion, the local rate of temperature increase is given by the SAR [24] for a given material. This extrapolation yields the maximum rate of local tissue temperature increase for a “hot spot”. Actual temperature increases will be less, due to thermal diffusion.

The view in Figure 9 shows a mid-plane saggital section of a Rhesus monkey head. Figure 9(a) shows normalized SAR values for the rhesus monkey head, irradiated in MEHK orientation at



800 MHz, as calculated by the FDTD code. The local SAR value at voxel (56,40,28) is 378.2W/kg (see detail of Figure 9 for the voxel location). The voxel size is 2.5mm x 2.5mm x 2.5mm. Figure 9(b) shows the Phase I Code temperature profile, without diffusion, for the monkey head exposed for 30 seconds at a power density of 500 mW/cm<sup>2</sup>. Figure 9(c) shows the temperature distribution with diffusion. The initial temperature of the head is 32.5 °C for each test.



○ : mid-plane saggital view, detail of Voxel (56,40,28)

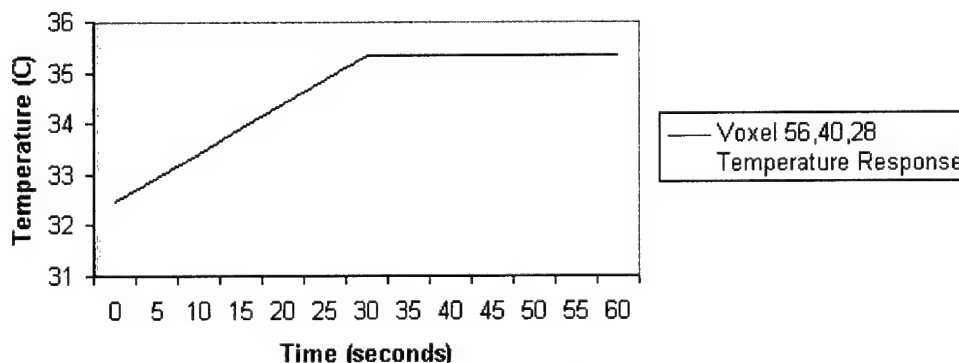
**Figure 9: 800MHz, 500mW/cm2, a) SAR, b) temp-no diffusion, c) temp-diffusion**

The Phase I Code prediction for voxel (56,40,28), without diffusion, is shown in Figure 10. Voxel (56,40,28) is a spinal region of the monkey carcass. This voxel reaches an equilibrium temperature of 35.3365 °C following 30 seconds of SAR exposure.

### Spine Hot Spot, Voxel 56,40,28

No Diffusion Test Case

SAR = 378.2 W/kg



**Figure 10: Adiabatic, No-Diffusion Temperature Response**



The heat equation for the no-diffusion test case simplifies to:

$$Q = mc_p \frac{dT}{dt}$$

where  $Q$  is the local SAR value,  $m$  is the voxel mass and  $c_p$  is the specific heat of the voxel material. The solution to this equation for voxel (56,40,28) gives a temperature increase of 2.8365 °C, - exactly equal to the Phase I Code prediction. The effects of diffusion will be discussed in Section 5.3.

### 5.3 Qualitative Comparison

A series of qualitative comparisons using voxel geometry of a rat were made. In each test case, the rat geometry consists of homogeneous tissue equivalent material (TEM) with the properties described in Table 4.

**Table 4: Tissue Equivalent Material properties**

TEM Properties	Value
$\rho$ , TEM mass density	1,000 kg/m <sup>3</sup>
$c_p$ , TEM specific heat	3,700 J/kgK
$k$ , TEM thermal conductivity	0.535 W/mK

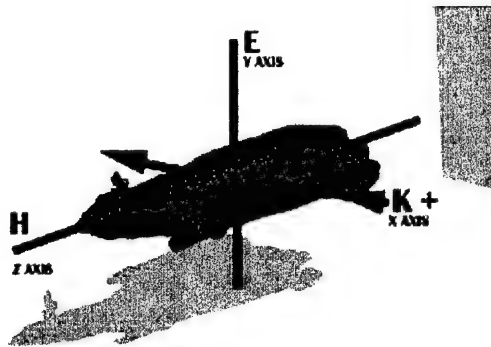
The purpose of these tests is to show that the Phase I Code provides reasonable and logical results under a variety of test cases. Table 5 describes the tests which were conducted. In this section, we discuss the results of these tests.

**Table 5: Qualitative Test Cases**

Test	Exposure Frequency (MHz)	Power Density (mW/cm <sup>2</sup> )	Exposure Time (seconds)	Boundary Conditions	Test Description
1	2060	100	30	adiabatic	Compare SAR to temperature distribution
2	2060	10/100	300/30	adiabatic	Compare effects of equal overall heating at different power densities
3	2060	10	300	convection/adiabatic	Comparison of convection vs adiabatic boundary condition
4	900/2060	100	30	adiabatic	Comparison of equal heating at different wavelengths

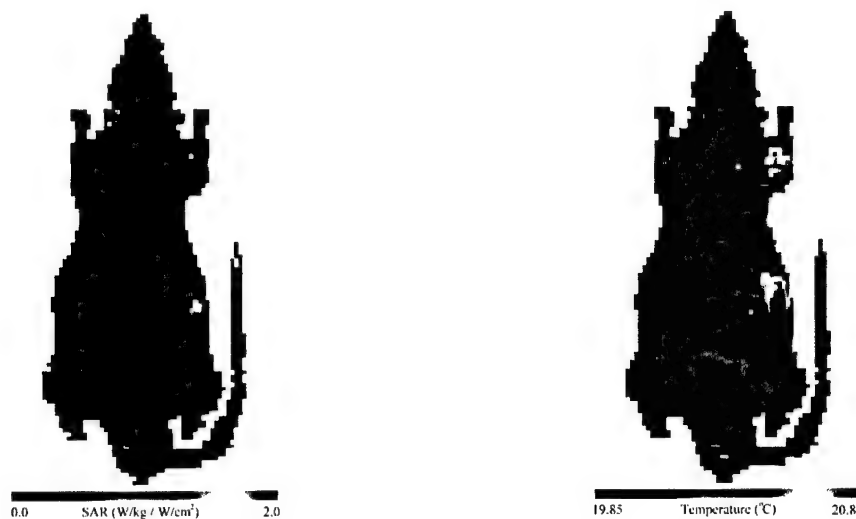
All results were obtained using a +HKE orientation for the transmitter. Figure 11 illustrates this orientation.





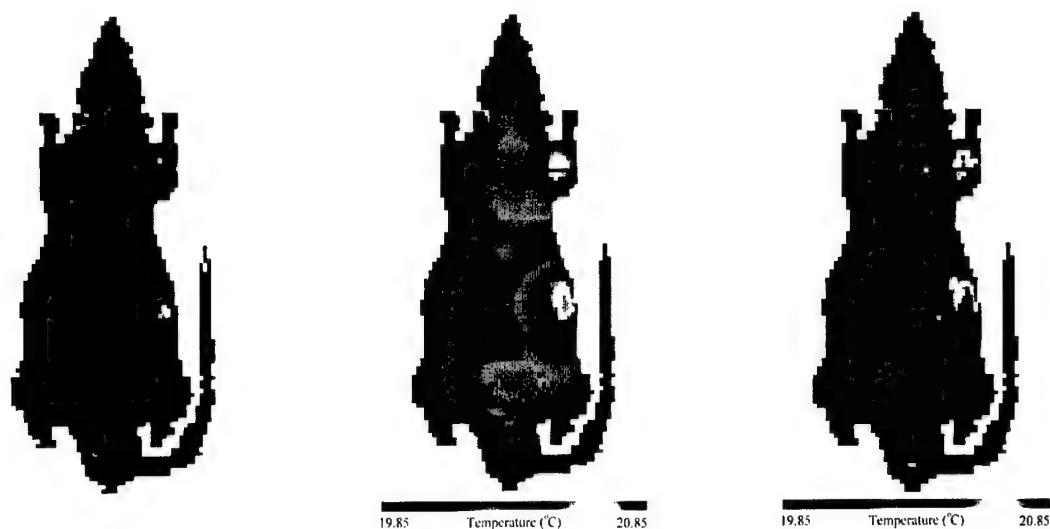
**Figure 11: +HKE orientation for the transmitter**

Figure 12 shows the results of Test 1. The purpose of Test 1 was to verify that the thermal response to SAR heating is appropriate. We would expect the temperature profile over the 3-D voxel geometry to approximately match the FDTD SAR distribution, as Figure 12 shows. Note that temperature diffusion, due to thermal conductivity and the thermal capacitance of the tissue material, tends to smooth the temperature distribution.

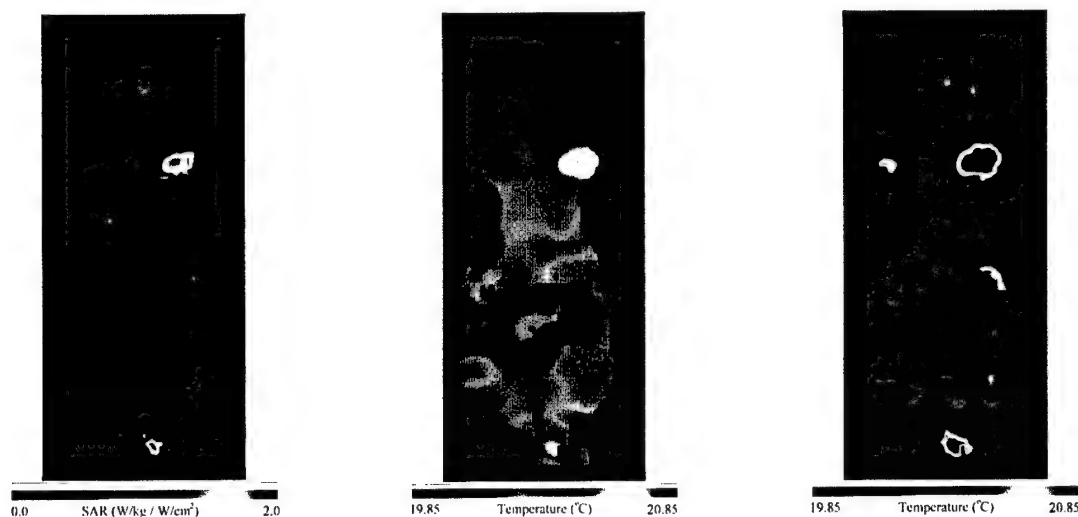


**Figure 12: Test 1, SAR heating (left), temperature response (right)**

Figure 13 shows the results from Test 2, a comparison with differing power densities and exposure times, but equal overall heating. Figure 14 features the same comparisons, shown in a plane view with the highest SAR values. The purpose of this test was to investigate the thermal effects of varying SAR intensities and exposure times. The test shows that a higher power density for a shorter exposure time results in higher peak temperatures than a low power density extended over a longer period of time, even though the overall energy absorbed in each test is equal.

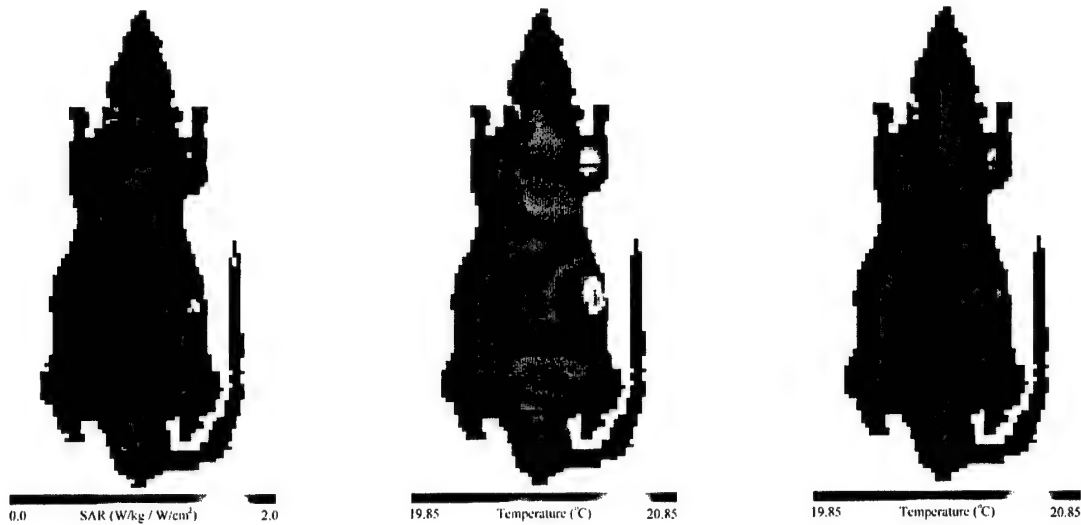


**Figure 13: Test 2, SAR distribution, low power density, high power density exposures**



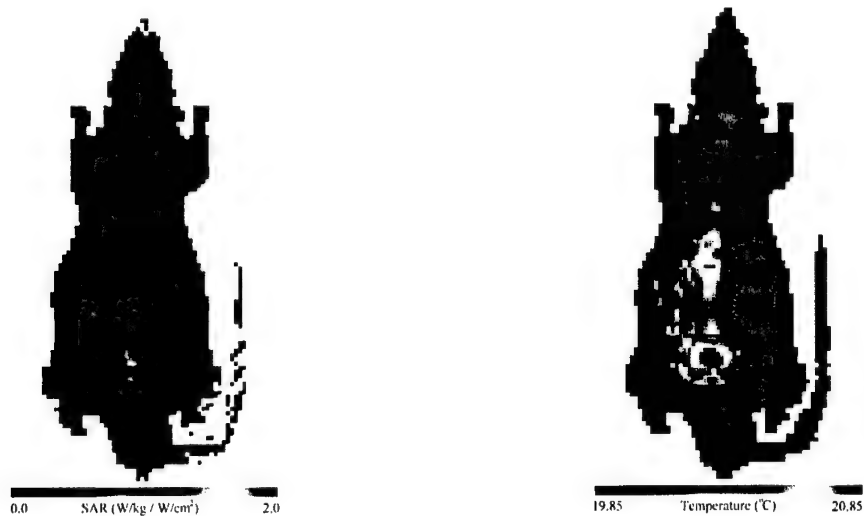
**Figure 14: Test 2, SAR distribution, low power density, high power density exposures**

Figure 15 shows the results of Test 3. This test explores how the thermal response to SAR heating differs between an adiabatic surface condition and a convection boundary condition. The convection boundary condition tends to decrease the SAR heating effects near the surface of the rat. SAR exposures that result in heating of centrally located sections of the rat will not be initially affected by this boundary condition because of the time lag associated with energy diffusion.



**Figure 15: Test 3, SAR distribution, adiabatic surface, convective boundary condition**

Figure 16 and Figure 17 show the results of Test 4. This test investigates the effects of SAR heating at differing wavelengths. Figure 16 depicts heating at cellular phone frequency, 900Mhz, while Figure 17 shows the effects of heating at microwave oven frequency, 2060Mhz. The results of this test are similar to those of Test 1, where the contours of the temperature response, though diffused, match the distribution of SAR heating. These images also highlight the effect that exposure frequency has on the distribution and intensity of heating.



**Figure 16: Test 4, SAR heating at 900Mhz, temperature response**

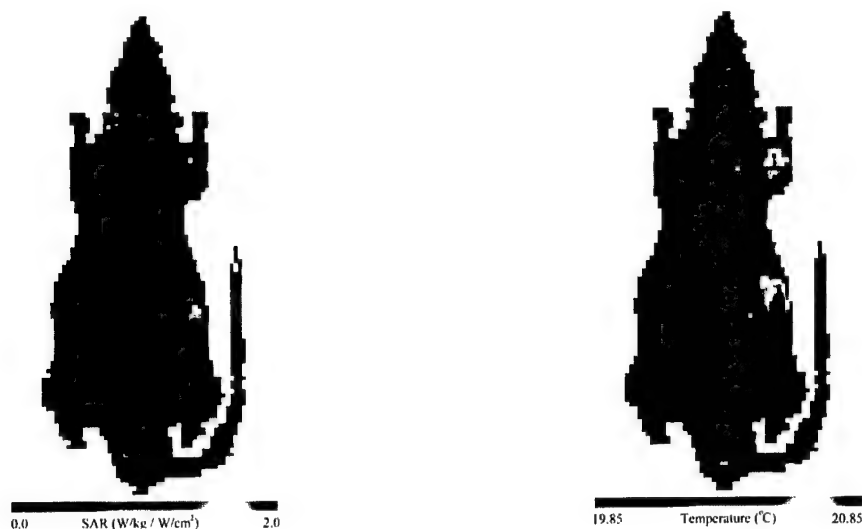


Figure 17: Test 4, SAR heating at 2060Mhz, temperature response

#### 5.4 Monkey Head Test Case Comparison

Probe measurements from the experiment described in Table 6 were compared against Phase I Code predictions for a Rhesus monkey carcass.

Table 6: Irradiated Monkey Test Case

Exposure Frequency (MHz)	Exposure Orientation	Exposure Power Density (mW/cm <sup>2</sup> )	Monkey Number	Comparison Location	Probe Track number	Exposure Duration (seconds)	Test Date
800	ME-HK	500	798Z	Spinal Hot Spot	1	30	April 18 2000

The FDTD code was used to predict locations within the carcass with high SAR heating. The temperature probe measurement locations shown in Figure 18 were determined using the FDTD results.

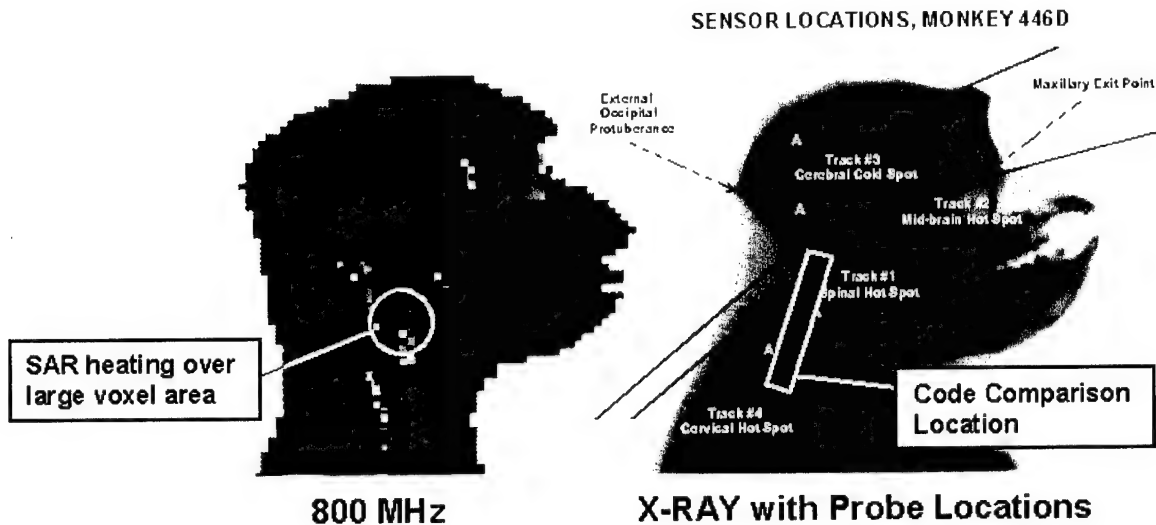


Figure 18: FDTD SAR prediction and probe locations

It is impossible to get exact agreement between predicted and measured thermal "hot spots," since the voxelized Rhesus monkey model used in this test to estimate SAR heating is not a model of the same monkey that was used to take probe measurements. Anatomical differences between the test subject and the model can be seen in Figure 18.

The Track 1 Spinal Hot Spot probe is located in a region where SAR heating is spread over a relatively large number of voxels. Consequently, this is the region where probe measurements are likely to agree with Phase I Code thermal predictions. Figure 19 shows a plot of spinal temperature measurements and Phase I Code predictions for the test case described in Table 6. Although we do not expect exact agreement, we do expect to capture the proper trend, which is the case.

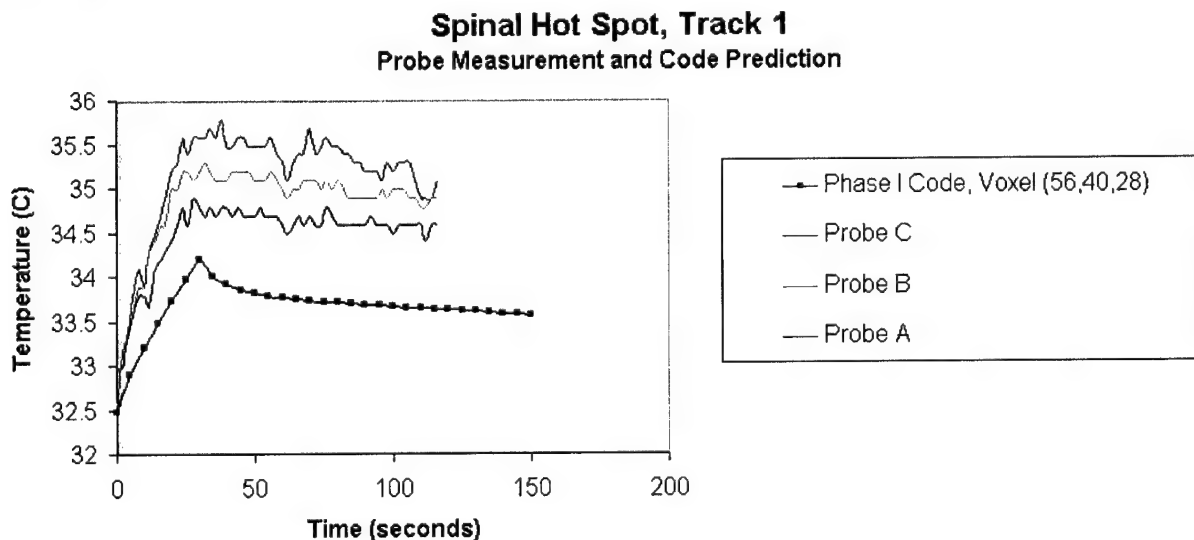


Figure 19: Probe measurements and Phase I Code prediction



Several factors affect the accuracy of the prediction for this test case, including;

- Anatomical differences between test subjects,
- Relatively large voxel size (2.5mm) of the model which can cause “hot spots” to be averaged out, and,
- Non-uniform initial temperature of the irradiated test subject.

We attribute the Phase I Code under-prediction in this test case to, primarily, the anatomical difference between test subject and the model.

## 5.5 Validation Summary

The Phase I Phase I Code has been tested:

- Against an analytical test case (sphere test), with near perfect agreement;
- Against a zero-diffusion analytical comparison, with perfect agreement;
- For conceptual validity under several scenarios using rat geometry; and,
- Against measured temperatures in an irradiated Rhesus monkey carcass.

Based on the results of these tests, we consider the Phase I Code to be an accurate prediction tool and a validated baseline upon which the more sophisticated Phase II thermal process modeling will be built.

## 6.0 RELEVANT PUBLICATIONS

1. Bernardi, P., Cavagnaro, M., Pisa, S., “SAR Distribution and Temperature Increase in an Anatomical Model of the Human Eye Exposed to the Field Radiated By the User Antenna in a Wireless LAN,” *IEEE Trans. Microwave Theory & Tech.*, vol.46, No. 12, December 1998.
2. Wissler, E., “Mathematical Simulation of Human Thermal Behavior Using Whole Body Models,” *Heat transfer in Medicine & Biology*, Vol. 1, 1985.
3. Bligh, J., “Temperature Regulation in Mammals and Other Vertebrates,” *Frontiers of Biology*, Vol. 30, North Holland Publishing, Amsterdam, 1973.
4. Chato, J.C., “Heat Transfer in Bioengineering,” *Advanced Heat Transfer*, B.T. Chao, ed. (University of Illinois Press, Urbana, 1969), p. 404-412.
5. Giering, K., Lamplecht, I., Minet, O., “Specific Heat Capacities of Human and Animal Tissues,” *SPIE* 2624, Sept, 1995, p. 188-197.
6. Takemori, T., Nakajima, T., Shoji, Y., “A Fundamental Model of the Human Thermal System for Prediction of Thermal Comfort,” *Heat Transfer – Japanese Research*, #24(2), 1995, p. 147-165.
7. Delhomme, G., Newman, W., Roussel, B., Jouvett, M., Bowman, H., Dittmar, A., “Thermal Diffusion and Instrument System for Tissue Blood Flow Measurements: Validation in Phantoms and *In Vivo* Organs,” *IEEE Transaction on Biomedical Engineering*, Vol. 41, No. 7, July, 1994.
8. Leonard, J.B., et al, “Thermal Properties of Tissue Equivalent Phantom Materials,” *IEEE Transactions on Biomedical Engineering*, Vol. BME-31, No. 7, July, 1984, p. 535.
9. Eugene Stanley, H., “Physics and Biomaterials Science,” MIT Press, Cambridge, MA, 1972, p. 231.
10. Shitzer, A., Eberhart, R., “Heat Generation, Storage, and Transfer Processes,” *Heat Transfer in Medicine and Biology*, Vol. 1, 1995, p. 137-152.



11. Pennes, H.H., "Analysis of Tissue and Arterial Blood Temperature in the Resting Forearm," *Journal of Applied Physiology*, Vol. 1, p. 93-122, 1948.
12. Spiegel, R.J., Fatmi, M.B.E., and Ward, T.R., "A Finite-Difference Electromagnetic Deposition/Phase I Codeulatory Model: Comparison Between Theory and Measurements," *Bioelectromagnetics*, 8:259-273 (1987).
13. Chandra, A. and Mukherjee, S., 1997, *Boundary Element Methods in Manufacturing*, Oxford University Press, New York (ISBN: 0-19-507921-3).
14. Curran, A.R., Editor, "User's Manual for PRISM 3.2," ThermoAnalytics Inc., Calumet, MI, May 1997.
15. Curran, A.R., et. al. "Automated Radiation Modeling for Vehicle Thermal Management," 1995 SAE International Congress & Exposition, Exhaust Systems & Shielding Session, Paper Number 950615, Detroit, MI, February 1995.
16. Johnson, K.R., et. al. "A Methodology for Rapid Calculation of Computational Thermal Models," 1995 SAE Congress - Underhood Thermal Management Session, Detroit, MI, February 1995.
17. Curran, A.R. et al, "Enhancements to the Vehicle Designer / PRISM Interface," Fifth Annual Ground Target Modeling and Validation Conference, Michigan Technological University, Houghton, Michigan, August 1994.
18. Johnson, K.R., et. al. "Thermal Modeling in Automotive Design," Proceedings of the Fifth Annual Ground Target Modeling & Validation Conference, Houghton, MI, August 1994.
19. Johnson, K. R., "Present State and Future of Infrared Signature Models," Proceedings of the Third Annual Ground Target Modeling and Validation Conference, MTU/KRC, August 1992.
20. WinTherm documentation, publications, and a fully functional but node-limited demo can be downloaded from <http://www.thermoanalytics.com>.
21. Nelson, D., M. Nelson, T. Walters, and P. Mason, "Thermal Effects of Millimeter Wave Irradiation of the Primate Head: Model Results" *IEEE Transactions on Microwave Theory and Techniques* (in press)
22. Mason, P., W. Hurt, J. D'Andrea, T. Walters, K. Ryan, P. Gajsek, D. Nelson, K. Smith and J. Ziriaux, "Effects of Frequency, Permittivity, and Voxel Size on Predicted Specific Absorption Rate Values during Electromagnetic Field Exposure" *IEEE Transactions on Microwave Theory and Techniques* (in press)
23. Mason, P., W. Hurt, J. Ziriaux, T. Walters, K. Ryan, D. Nelson and J. D'Andrea "Models used to Determine the Bioeffects of Directed Energy Exposure" in *Countering the Directed Energy Threat: Are Closed Cockpits the Ultimate Answer?* NATO Research and Technology Organization report RTO-MP-30 (2000)
24. Walters, T. J., P. A. Mason, K. L. Ryan, D. A. Nelson, and W. D. Hurt "A Comparison of SAR Values Determined Empirically and by FD-TD Modeling" In: B.J. Klauenberg and D. Miklavic (Eds.), *Radio Frequency Radiation Dosimetry*, Dordrecht, The Netherlands: Kluwer Academic Publishers (1999)
25. Nelson, D., Walters, T., Ryan, K. and Johnson, L.R. "Skin Heating Effects of Millimeter Waves: Inter-Species Variability" *Advances in Heat Transfer and Mass Transfer in Biotechnology 1999*, ASME HTD-Vol. 363; BED-Vol. 44, pp. 141-4 (1999)
26. Nelson, D. "Invited Editorial on 'Pennes' 1948 Paper Revisited" *Journal of Applied Physiology* 85:2-3 (1998)





27. Durney, C.H., Massoudi, H., and Iskander, M.F., (1986) *RadioFrequency Radiation Dosimetry Handbook*, USAF School of Aerospace Medicine Report, USAFSAM-TR-85-73, Brooks Air Force Base, TX., Fourth Edition.
28. IEEE Standard for Safety Levels with Respect to Human Exposure to Radio Frequency Electromagnetic Fields, 3kHz to 300 Ghz: IEEE Standard C95.1, 1999 Edition. (1999).
29. Kolka, M. A. and Elizondo, R.S. "Thermoregulation in *Erythrocebus patas*: a thermal balance study" J.Appl.Physiol. 55:1603-1608 (1983).
30. Adair, ER and Adams, BW "Adjustments in metabolic heat production by squirrel monkeys exposed to microwaves" J. Applied Physiology 52:1049-1058, 1982].
31. Huiskes, R., Janssen, J.D., and Slooff, T.J., 1982, "*Finite element analysis for artificial joint fixation problems in orthopaedics.*" Finite Elements in Biomechanics, RH Gallager, BR Simon, PC Johnson, and JF Gross, eds. (New York: J. Wiley).
32. Nelson, D., M. Nelson, T. Walters, and P. Mason, "*Thermal Effects of Millimeter Wave Irradiation of the Primate Head: Model Results*" IEEE Transactions on Microwave Theory and Techniques (in press).
33. Nelson, D., Walters, T., Ryan, K. and Johnson, L.R. "*Skin Heating Effects of Millimeter Waves: Inter-Species Variability*" Advances in Heat Transfer and Mass Transfer in Biotechnology 1999, ASME HTD-Vol. 363; BED-Vol. 44, pp. 141-4 (1999).
34. Incropera, F and DeWitt, D., "Fundamentals of Heat and Mass Transfer, Fourth Edition," John Wiley and Sons, New York, 1996.
35. Carslaw, H.S., and Jaeger, J.C., "Conduction of Heat in Solids, 2d ed." Oxford University Press, 1959 p. 492.



## Appendix A. Tissue Properties

Tissue Type	Species	$\rho$ [kg/m <sup>3</sup> ]	$C_p$ [J/kgK]	$k$ [W/mC]	Blood Flow [L/s kg]	$\alpha$ [m <sup>2</sup> /s]	O <sub>2</sub> Consum [L/kg s]
Air (external)	Human						
Air (internal)	Human						
Bile	Human	1010					
Bladder	Human	1040					
Blood (whole)	Human	1060	3930	0.488		1.17E-07	
Blood Vessels	Human						
Aorta	Human			0.052		1.27E-07	
Aorta	Human					1.56E-07	
Aorta	Human					1.22E-07	
Artery	Human	1062.5					
Vein	Human	1056					
Body Fluid	Human						
Bile, liver	Human	1010					
Cerebrospinal fluid	Human	1007					
Gastric juice	Human	1007					
Saliva	Human	1007					
Sweat	Human	1004.5					
Synovial Fluid	Human	1011.5					
Bone	Human	1990	1300	0.3	0	1.16E-07	
Bone marrow (red)	Human	1027					
Bone marrow (yellow)	Human	980					
Bone marrow	Pig		2740				
Brain white matter	Human	1043	3600	0.503		1.38E-07	
Brain grey matter	Human	1039	3680	0.566		1.34E-07	
Brain cerebellum	Human	1040	3500	0.6		1.47E-07	
Brain (whole)	Human				0.00965		
Brain (whole)	Rat				0.0232		
Cartilage	Human	1098	2300	0.22		8.71E-08	
Eye (cornea)	Human	1076					
Eye(lens)	Human	1077.5	3900	0.5		1.19E-07	
Eye (retina)	Human						
Eye (sclera)	Human	1010	3900	0.5		1.27E-07	
Eye (aqueous humor)	Human	1003					
Fat	Human	916	2300	0.204	0.000004333	1.31E-07	
Fat	Rat				0.00425		
Gall Bladder	Human						
Glands	Human						
Heart	Human	1060	3720	0.587		1.48E-07	
Intestine (Large)	Human	1044					
Intestine (small)	Human	1044					
Kidney	Human	1050	3890	0.5385		1.32E-07	



Kidney	Rat				0.151		
Kidney cortex	Human	1049		0.547		1.47E-07	
Kidney Medula	Human	1044		0.54		1.48E-07	
Kidney Pelvis	Human					1.37E-07	
Ligaments	Human						
Liver	Human	1050	3600				60
Liver	Rat			0.572	0.0212		
Lung	Human	1050		0.518	0.00045		
Lymph	Human	1030					
Mucous Membrane	Human						
Muscle	Human	1041	3500	0.6	0.000516667	1.65E-07	0.000338
Muscle	Rat				0.0047		
Nails	Human	1300					
Nerve	Human	1038					
Pancreas	Human	1045		0.345			
Skin/ dermis	Human	1116	3500	0.545	0.00395	1.4E-07	
Spleen	Human	1054	3720	0.545		1.44E-07	
Spleen	Rat				0.06167		
Stomach	Human	1050		0.527			60
Stomach	Rat				0.0275		
Testicles	Human	1044					
Tooth	Human	2165	1170	0.45		1.78E-07	



## Appendix B.1 Class Definitions for the Solver

```
/*!
 * \file threedmesh.h
 * \brief Definition of ThreeDMesh class
 *
 * This file contains the definition of the ThreeDMesh class, which
 * stores the 3-D voxelized mesh needed for the thermoregulation
 * thermal solver
 */
/*-----
 Copyright (C) 2001 ThermoAnalytics, Inc.
-----*/

#ifndef TAI_ThreeDMesh_H
#define TAI_ThreeDMesh_H

/*-----
 standard includes
-----*/

/*-----
 non-standard includes
-----*/

#include "base/taisys.h"
#include "taicpp/vector.h"
#include "dstruct/threedmeshsize.h"

/*-----
 public #defines, enums, etc.
-----*/

/*-----
 public typedefs, class definitions, etc.
-----*/

/*!
 * \brief Data class containing data for each cell.
 *
 * The ThreeDMesh class contains a vector of ThreeDCell classes.
 *
 * \sa ThreeDMesh
 */
class ThreeDCell {

public:
    ThreeDCell();
    ThreeDCell(const ThreeDCell &);
    ~ThreeDCell();

    ThreeDCell& operator=(const ThreeDCell &);

    float T() const { return vT; }
    float Told() const { return vTold; }
    int tissueIndex() const { return vTissueIndex; }
    float Qsar() const { return vQsar; }

    void setT(float T) { vT = T; }
    void setTold(float T) { vTold = T; }
    void setTissueIndex(int idx) { vTissueIndex = idx; }
    void setQsar(float Q) { vQsar = Q; }
```



```
private:
    float vT;           // Temperature of cell.
    float vTold;        // Temperature of cell at last time step.
    int vTissueIndex;   // The tissue the cell is made up of.
    float vQsar;        // The energy absorbed by the cell from RF sources

};

/*!
 * \brief Data class for storing the 3-D voxelized mesh information.
 *
 * This data class stores the size of the mesh (number of cells in
 * x, y, z), the cell dimension (x, y, z) and a vector of ThreeDCell
 * data classes. (one instance of ThreeDCell for each cell in the mesh)
 *
 * \sa ThreeDCell
 */
class ThreeDMesh{
public:

    ThreeDMesh();
    ThreeDMesh(ThreeDMeshSize &meshSize);
    ~ThreeDMesh();

    int resize(ThreeDMeshSize &meshSize);

    int nCells() const { return vMeshSize.nCells(); }
    int nCellX() const { return vMeshSize.nCellX(); }
    int nCellY() const { return vMeshSize.nCellY(); }
    int nCellZ() const { return vMeshSize.nCellZ(); }

    float cellSizeX() const { return vMeshSize.cellSizeX(); }
    float cellSizeY() const { return vMeshSize.cellSizeY(); }
    float cellSizeZ() const { return vMeshSize.cellSizeZ(); }

    const ThreeDCell& operator()(int i, int j, int k) const;
    ThreeDCell& operator()(int i, int j, int k);

    const ThreeDCell& operator()(int idx) const { return vCell[idx]; }
    ThreeDCell& operator()(int idx) { return vCell[idx]; }

    bool validIndex(int i, int j, int k) const;

#ifdef HAS_NAMESPACES
    friend std::ostream& operator<<(std::ostream &s,
                                    const ThreeDMesh &data);
#else
    friend ostream& operator<<(ostream &s, const ThreeDMesh &data);
#endif

    void printSize();

private:

    ThreeDMeshSize vMeshSize; // The size and dimensions of the mesh.
    vector <ThreeDCell> vCell; // Holds a ThreeDCell instance for each cell.

private:
    /*! Disabled copy constructor */
    ThreeDMesh(const ThreeDMesh &);
    /*! Disabled assignment operator */
    ThreeDMesh& operator=(const ThreeDMesh &);
```



```
};

/*-----
public function prototypes
-----*/

/*-----
inline function definitions
-----*/

/*-----*/
/*!
/*!
* \brief Retrieves the cell at the specified i,j,k location.
*
* \return Returns a const reference to the cell.
* \sa
*/
inline const ThreeDCell& ThreeDMesh::operator()(int i, int j, int k) const
{
    return vCell[(k*nCellY()+j)*nCellX()+i];
}

/*-----*/
/*!
* \brief Retrieves the cell at the specified i,j,k location.
*
* \return Returns a non-const reference to the cell.
* \sa
*/
inline ThreeDCell& ThreeDMesh::operator()(int i, int j, int k)
{
    return vCell[(k*nCellY()+j)*nCellX()+i];
}

/*-----*/
/*!
* \brief Checks if the specified i, j, k indices are valid indices
*        for the mesh.
*
* \param i Index in X direction. (should be >= 0 and < nCellX())
* \param j Index in Y direction. (should be >= 0 and < nCellY())
* \param k Index in Z direction. (should be >= 0 and < nCellZ())
*
* \return Returns true if the indices are valid
* \sa
*/
inline bool ThreeDMesh::validIndex(int i, int j, int k) const
{
    if ( i >= 0 && i < nCellX() &&
        j >= 0 && j < nCellY() &&
        k >= 0 && k < nCellZ() ){
        return true;
    }
    else{
        return false;
    }
}

#endif /* end #ifndef TAI_ThreeDMesh_H */
/*-----
End of file - don't put any code beyond the above #endif
-----*/
```



## Appendix B.2 Solver Source Code

```
/*!  
* \file threedsolver.cpp  
* \brief Implementation of the Solve3DMesh class. This class is  
*        used for exercising the thermal solution of voxelized  
*        domain.  
*  
* The Solve3DMesh class is intended to be used along with the  
* main thermal solver. The main thermal solver has functions  
* for adding multiple instances of the Solve3DMesh class, each  
* of which would represent a different voxelized domain. The  
* Solve3DMesh class only has the ability to exercise a single  
* iteration of its relaxation numerical solution. All of the  
* looping (iteration and time stepping) is handled by the main  
* thermal solver.  
*  
* The Solve3DMesh class uses Crank-Nicholson and SOR.  
*  
*/  
/*-----  
Copyright (C) 2001 ThermoAnalytics, Inc.  
-----*/  
  
/*-----  
associated include  
-----*/  
#include "threedsolver.h"  
  
/*-----  
standard includes  
-----*/  
#include <math.h>  
/*-----  
non-standard includes  
-----*/  
#include "base/basic.h"  
#include "dstruct/matldata.h"  
  
/*-----  
private #defines, enums, etc.  
-----*/  
  
/*-----  
private typedefs, class definitions, etc.  
-----*/  
  
/*-----  
private (static) global variables (used sparingly)  
-----*/  
  
/*-----  
private (static) function prototypes  
-----*/  
  
/*-----*/  
/*!  
* \brief Constructor for Solve3DMesh  
*  

```





```
* \param mesh reference to the voxel domain
* \param material pointer to material database
*
* \sa ~Solve3DMesh()
*/
Solve3DMesh::Solve3DMesh(ThreeDMesh &mesh, MaterialData *material)
{
    vMesh = &mesh;

    material->retrieveConductivities(vConductivity);
    material->retrievePropertyForAllMaterials(Material::SpecificHeat,
                                              vSpecificHeat);
    material->retrievePropertyForAllMaterials(Material::Density, vDensity);

    vDx = mesh.cellSizeX();
    vDy = mesh.cellSizeY();
    vDz = mesh.cellSizeZ();

    vDx2 = vDx*vDx;
    vDy2 = vDy*vDy;
    vDz2 = vDz*vDz;

    vNCellX = mesh.nCellX();
    vNCellY = mesh.nCellY();
    vNCellZ = mesh.nCellZ();

    vTimeStep = 1.0;
    vRealTime = 0.0;
    vDuration = 10.0;
}

/*-----*/
/*!
 * \brief Destructor for Solve3DMesh
 *
 * \sa Solve3DMesh(ThreeDMesh &mesh, MaterialData *material)
 */
Solve3DMesh::~Solve3DMesh()
{
    // nothing to do for now.
}

/*-----
 public function definitions
-----*/

/*-----*/
/*!
 * \brief Performs a single iteration of the thermal solution.
 *
 * \param maxDeltaT The maximum change in cell temperature (deg C)
 *                  for this iteration. (computed by this function)
 *
 * \return Returns SUCCESS upon successful completion.
 *         Other return values indicate a failure occurred.
 * \sa
 */
int Solve3DMesh::iterateOnce(float &maxDeltaT)
{
    int status = SUCCESS;

    bool steadyState = false; //-- not allowing steady state solution for now.
```



```
ThreeDMesh &curMesh = mesh();

const int nNeighborCells = 6;
ThreeDCell *neiCellPtrList[nNeighborCells];

const float neiCellDimension[nNeighborCells] =
    { dx(), dx(), dy(), dy(), dz(), dz() };

const float neiCellDimensionSquared[nNeighborCells] =
    { dx2(), dx2(), dy2(), dy2(), dz2(), dz2() };

float deltaT;    // Change in temperature for a node for this iteration
maxDeltaT = -1;  /* reset the maximum observed change in temperature
                  * in any node during this iteration. */

float absDeltaT;    // Absolute value of temperature change.

// Loop over entire domain and solve for the temperature of each cell
for (int k = 0; k < nCellZ(); k++) {
    for (int j = 0; j < nCellY(); j++) {
        for (int i = 0; i < nCellX(); i++) {
            ThreeDCell &cell = curMesh(i, j, k);

            int tissueIdx = cell.tissueIndex();

            static const int externalAir = 0;  //-- temporary define
            static const int internalAir = -1; //-- temporary define

            static float hVal = 10.0;  //-- temporary value
            static float externalAirTemp = 293.15;  //-- temporary value

            if (tissueIdx != externalAir) { /* we don't compute anything for
                                             * external air cells. */
                float kCurCell = conductivity(tissueIdx);

                float condSum = 0.0;
                float condTSum = 0.0;
                float condToldSum = 0.0;

                // Create an array that contains pointers to each neighboring cell.
                int n;
                for (n = 0; n < nNeighborCells; n++) neiCellPtrList[n] = 0;
                if ( (i+1) < vNCellX ) neiCellPtrList[0] = &curMesh(i+1, j, k);
                if ( (i-1) >= 0 ) neiCellPtrList[1] = &curMesh(i-1, j, k);
                if ( (j+1) < vNCellY ) neiCellPtrList[2] = &curMesh(i, j+1, k);
                if ( (j-1) >= 0 ) neiCellPtrList[3] = &curMesh(i, j-1, k);
                if ( (k+1) < vNCellZ ) neiCellPtrList[4] = &curMesh(i, j, k+1);
                if ( (k-1) >= 0 ) neiCellPtrList[5] = &curMesh(i, j, k-1);

                // Loop for each neighboring cell and add conduction heat transfer
                // contributions from each neighbor. If there is no neighbor
                // on a cell face, add convection heat transfer with the ambient.
                for (n = 0; n < nNeighborCells; n++) {
                    ThreeDCell *neighborCell = neiCellPtrList[n];

                    if (neighborCell) {
                        int neiTissue = neighborCell->tissueIndex();

                        if (neiTissue == externalAir) {
                            // Apply convection if the neighbor is external air.
                            float conductor = hVal/neiCellDimension[n];
```



```
condSum      += conductor;
condTSum      += conductor*externalAirTemp;
condToldSum += conductor*externalAirTemp; /*/-- modify if
                                         * external air
                                         * changes
                                         * with time. */
}
else if (neiTissue == internalAir) {
    /*/-- do we do anything special with internal air?
}
else{

    // Apply conduction.  If the neighboring cell has a different
    // tissue, an equivalent conductivity must be calculated.
    float kVal = 0;
    if (neiTissue != tissueIdx) {
        float kNei = conductivity(neiTissue);
        kVal = 2*kCurCell*kNei/(kCurCell+kNei);
    }
    else{
        kVal = kCurCell;
    }

    float conductor = kVal/neiCellDimensionSquared[n];

    condSum      += conductor;
    condTSum      += conductor*neighborCell->T();
    condToldSum += conductor*neighborCell->Told();
}
}

// Add contributions due to heat transfer with the blood in this cell.
float mDotBlood = 0.034; /*/-- temporary value.
float CpBlood   = 0;      /*/-- temporary value.
float Tblood    = 311.0;  /*/-- temporary value.
float TbloodOld = 311.0;  /*/-- temporary value.

float conductor = mDotBlood*CpBlood;

condSum      += conductor;
condTSum      += conductor*Tblood;
condToldSum += conductor*TbloodOld;

// Sum Q's (sar and metabolic heating).  Need to multiply by
// the density since the Q's are stored in W/kg.
float rho = density(tissueIdx);

float q = rho*(cell.Qsar() + metabolicHeating(tissueIdx));

// Evaluate and store new temperature for this cell.
float Tlast = cell.T();
float T;
if (steadyState){
    T = (condTSum + q)/condSum;
}
else{
    float Cp = specificHeat(tissueIdx);
    float tranFactor = 2*rho*Cp/timeStep();
    float Told = cell.Told();

    T = (condTSum + condToldSum -
        Told*(condSum-tranFactor) + 2.0f*q)/
```



```
(condSum+tranFactor);
}

deltaT = T - Tlast;

T = Tlast + relax()*deltaT;

absDeltaT = fabs(deltaT);
maxDeltaT = Max(absDeltaT, maxDeltaT);

cell.setT(T);
}
}
}

return status;
}

/*-----*/
/*!
 * \brief Writes the thermal results to stdout for testing/debugging
 * purposes.
 */
void Solve3DMesh::reportResults() const
{
    const ThreeDMesh &curMesh = mesh();

    cout << "#Real Time = " << realTime() << " (seconds)\n";
    cout << curMesh;
}

/*-----*/
/*!
 * \brief Copies new temperature to old temperatures
 *
 * This function is needed to move new cell temperatures to old cell
 * temperatures after a time step is complete. This function is called
 * by the main thermal solver at the appropriate time.
 *
 * \sa iterateOnce()
 */
void Solve3DMesh::copyNewToOld()
{
    ThreeDMesh &curMesh = mesh();

    int idx;
    for (idx = 0; idx < curMesh.nCells(); idx++) {
        ThreeDCell &cell = curMesh(idx);
        cell.setTold(cell.T());
    }
}

/*-----
public SLOT function definitions
-----*/

/*-----
protected function definitions
-----*/
```



```
/*-----  
private SLOT function definitions  
-----*/  
  
/*-----  
private function definitions  
-----*/  
  
/*-----  
End of file  
-----*/
```



## Appendix B.3 Sample Input File

```
# Input file for thermoreg thermal solver.

#####
# Voxel description
#####
# Specify the number of voxels in X, Y, and Z directions
NumberOfVoxelsInX: 51
NumberOfVoxelsInY: 22
NumberOfVoxelsInZ: 114

# Specify voxel sizes in mm
VoxelSizeInX: 1.95
VoxelSizeInY: 1.95
VoxelSizeInZ: 2.15

#####
# Numerical solution parameters
#####

# Specify the duration of exposure to radiation in seconds.
# (The radiation exposure will be applied from time zero and
# will drop to zero after the specified time has elapsed.)
DurationOfExposure: 30

# Specify the duration of the analysis in seconds.
Duration: 30

# Specify the power density of the radiation in milliwatts/cm^2
PowerDensity: 100

# Specify the time step in seconds.
TimeStep: 5

# Specify the initial temperature of the object in deg C
InitialObjectTemperature: 20

# Specify the ambient temperature in deg C
AmbientTemperature: 20

# Specify the surface convection coefficient in W/m^2-K
SurfaceConvectionCoefficient: 10

# Specify the solution relaxation value. (must be > 0 and < 2.0)
# Typically a value in the range of 1.0 to 1.8 would be used.
SolutionRelaxationValue: 1.5
```